確率的分散アルゴリズムにおける初期条件の階層について(計算量理論とその応用)

坂本, 直志

数理解析研究所講究録
第796巻 1992年 54-66

確率的分散アルゴリズムにおける初期条件の階層について

坂本 直志
Naoshi Sakamoto
東京工業大学 理学部 情報科学科

概要

分散アルゴリズムにおいて、各 processor に与える初期条件は重要である。

初期条件により、分散アルゴリズムのネットワーク上での、そのネットワークに対する計算能力が異なる。本稿では、同期した確率計算モデルにおいて、全頂点数がわかれば、または、1 つだけ異なる情報をもらうことにより、対称性が崩れた時には、ネットワークに対するあらゆる情報を得ることができ、また、単に全頂点が着色されていることだけでは、そのネットワークのグラフについて何も情報を得ていないことを示し、さらに、それらの間に計算能力の差があることを示した。

1 はじめに

分散アルゴリズムや並列アルゴリズムの設計において、従来、様々な初期条件が用いられてきた。

例えば、「各 processor に連続した番号が振られている。」「各 processor は、ネットワークの構造を知っている。」「ネットワークの全頂点数を知っている。」「頂点に色付けされている。」などである。

しかし、この条件を用いる根拠は、アルゴリズムの設計がうまくいくというほかになく、本当にその条件が必要であるかは、あまり議論されていなかった。

本論文では、従来用いられてきたような、一般的で代表的な初期条件について相対的な比較を行なった。2 章で示した、同期した確率的分散モデルについて、3 章で定義した初期条件についての相対的な複雑さを、4 章では決定性、5 章では確率的なモデルについて示した。

2 準備

$\mathbf{G}$ を有限連結無向グラフのクラスとする。$G \in \mathbf{G}$ について、$G = (V, E), |V| = N$ とする。(V は、有限集合、 $E \subseteq V \times V, E \neq \emptyset$)

$\mathbf{Z}_d = \{1, 2, \ldots, d\}$ とする。

$\deg(v)$ を頂点 v の次数とする。G の port numbering $F_G = \{f_v \mid v \in V\}$ とは、各 $v \in V$ について、$\mathbf{Z}_{\deg(v)}$ から v と隣接する頂点への全単射関数 f_v の集合とする。

アルゴリズム M とは、次の条件を満たすような Turing machine である。

1. テープは、入力用、作業用、通信用の入力、出力の組により構成され、さらに出力用のテープを持つ。

2. 有限状態制御部は、現在読めるテープのマスの内容と状態から、次の状態とテープのマスの内容とテープのヘッドの動きを決める。次の状態への遷移は高々 2 通りあり、2 通りあった時は、等確率に選び、遷移する。

このようなアルゴリズムを**確率的**と言い、有限状態制御部について、全ての遷移が、高々 1 通りしかないものを**決定性**と言う。

network $(G, F_G, M)(\mathbf{x}) = \mathbf{y}$ とは、「各 G の頂点上に Turing machine M が配され、各頂点 v 上の machine M_v には、$\mathbf{x} = \langle x_1, \ldots, x_N \rangle$ の中の x_v と、deg(v) が、入力テープに書き込まれる。各 M は、同期しながら動作する。M_v が通信用テープに $u\#m$ を書き、特別な状態 (送信状態) に入ると、$M_{f_v(u)}$ の受信用のテープに $f_{f_v(u)}(w) = v$ となる w について、$w\#m$ が書き込まれる。そして、全 processor が停止した時、出力テープに書かれている内容 $\langle y_1, \ldots, y_N \rangle = \mathbf{y}$ を出力とする。」ということを意味している。

問題 Q について、predicate $P_Q(G, \mathbf{x})$ が定義されている時、M が、問題 Q を解くとは、

$$(\forall G)(\forall F_G) \Pr[P_Q(G, (G, F_G, M)(\lambda))] > \frac{1}{2}$$

を満たすことである。

また、partial recursive function $\eta(G, F_G) = \mathbf{x}$ を初期値関数と言い、M が初期条件 η のもとで問題 Q を解くとは、

$$(\forall G)(\forall F_G) \Pr[P_Q(G, (G, F_G, M)(\eta(G, F_G)))] > \frac{1}{2}$$

を満たすことである。

time$((G, F_G, M)(\mathbf{x}))$ をこの network の実行時間とし、(確率的な値をとる) 確率変数 $T = \text{time}((G, F_G, M)(\mathbf{x}))$ について、Ex$[T]$ を平均実行時間と言い、初期条件 η のもとで M の平均実行時間が $p(N)$ であるとは、

$$(\forall G \text{ s.t. } G = (V, E),\ |V| = N)(\forall F_G)[\text{Ex}[\text{time}((G, F_G, M)(\eta(G, F_G)))] \leq p(N)]$$

を満たすことである。

初期条件 η のもとで M が誤りなしで問題 Q を解くとは、

$$(\forall G)(\forall F_G)[\Pr[P_Q(G,(G,F_G,M)(\eta(G,F_G)))]=1$$

$$\wedge\ \mathrm{Ex}[\mathrm{time}((G,F_G,M)(\eta(G,F_G)))]<\infty]$$

を満たすことである。

問題 A の pridicate を P_A、問題 B の pridicate を P_B とするとき、$A \leq_{\mathrm{m}} B$ via M であるとは、ある決定性アルゴリズム M が存在し、

$$(\forall G)(\forall F_G)[P_B(G,\mathrm{x}) \Rightarrow P_A(G,(G,F_G,M)(\mathrm{x}))]$$

を満たすことである。

$A \leq_{\mathrm{rm}} B$ via M であるとは、ある確率的アルゴリズム M が存在し、

$$(\forall G)(\forall F_G)[P_B(G,\mathrm{x}) \Rightarrow \Pr[P_A(G,(G,F_G,M)(\mathrm{x}))]>\frac{1}{2}]$$

を満たすことである。

$A \leq_{\mathrm{zpm}} B$ via M であるとは、ある確率的アルゴリズム M が存在し、

$$(\forall G)(\forall F_G)[[P_B(G,\mathrm{x}) \Rightarrow [\Pr[P_A(G,(G,F_G,M)(\mathrm{x}))]=1]\ \wedge$$

$$[\mathrm{Ex}[\mathrm{time}((G,F_G,M)(\eta(G,F_G)))]<\infty]]]$$

を満たすことである。

各 $r \in \{\mathrm{m},\mathrm{rm},\mathrm{zpm}\}$ について、$<_r, \equiv_r, |_r$ を次の様に定義する。

$$A \equiv_r B \quad \Leftrightarrow \quad A \leq_r B \wedge B \leq_r A$$

$$A <_r B \quad \Leftrightarrow \quad A \leq_r B \wedge B \not\leq_r A$$

$$A |_r B \quad \Leftrightarrow \quad A \not\leq_r B \wedge B \not\leq_r A$$

3 取り扱う問題について

グラフ G , $\mathbf{x} = \{x_1, \ldots, x_N\}$ について、

- $\emptyset$

  $P_{\emptyset}(G, \mathbf{x}) \Leftrightarrow x_i$ は、各頂点の次数の列。

- GRAPH-SIZE

  $P_{\text{GRAPH-SIZE}}(G, \mathbf{x}) \Leftrightarrow$ 全ての i について、 $x_i = N$

- EDGE-COLORING

  $P_{\text{EDGE-COLORING}}(G, \mathbf{x}) \Leftrightarrow \mathbf{x}$ は、1 つの頂点に入射する辺の着色が全て異なるような辺着色の表現。

- COLORING

  $P_{\text{COLORING}}(G, \mathbf{x}) \Leftrightarrow \mathbf{x}$ は、隣接する頂点全てと異なる頂点の着色の表現。

- LEADER

  $P_{\text{LEADER}}(G, \mathbf{x}) \Leftrightarrow$ ただ 1 つ $x_i = 1$ となる i があり、$i \neq j$ となる j については $x_j = 2$.

- NUMBERING

  $P_{\text{NUMBERING}}(G, \mathbf{x}) \Leftrightarrow \mathbf{x}$ は、 $\mathbf{Z}_N$ の permutation.

そのほかにも、これらの組合せにより、GRAPH-SIZE & EDGE-COLORINGや、GRARPH-SIZE & COLORINGなどが考えられる。

4 決定性の reduction $\leq_{\mathrm{m}}$ と完全性について

Lemma 4.1 *Leader* が決まっていれば、生成木が作れる。

Proof. 次のアルゴリズムを考える。

- leader である場合

  I. 隣接している頂点全てに 0 を送る。

  II. 隣接している頂点全てから 0 を受け取ったら停止する。

- leader でない場合

  i) 隣接している頂点から 0 が送られてくるのを待つ。

  ii) 一番始めに受け付けた 0 を送ってきた頂点を "親" とする。

  iii) 親以外の全ての頂点 "子" に 0 を送る。

  iv) 全ての子から 0 が送られてきたか、子が存在しない時は、親に 0 を送り停止する。

U_t を、時刻 t に状態 i) に入っている processor の集合、$R_t = V - U_t$ を、時刻 t に状態 I. II. ii) iii) iv) に入っている processor の集合、∂R_t を、時刻 t に状態 I. ii) iii) に入っている processor の集合、∂U_t を、U_t の元で ∂R_t の元と隣接している processor の集合とする。

$U_0 = V - \{\text{leader}\}$, $R_0 = \partial R_0 = \{\text{leader}\}$ である。また、$\partial U_t \subseteq U_t$, $\partial R_t \subseteq R_t$.

Claim 1 $(\forall t)(\exists t' \ s.t.\ (t < t'))[\partial U_t \subseteq R_{t'}]$

Proof of claim. ∂R_t に含まれる processor は、親以外の隣接している processor に 0 を送る。つまり、∂R_t の processor は、∂U_t の processor に 0 を送るので、ある時刻 t' には、∂U_t の全ての processor は、0 を受け取り、状態 i) を抜け出す。 △

Claim 2 $U_t \neq \emptyset \Rightarrow (\exists t' \ s.t. \ t' \leq t)[\partial U_{t'} \neq \emptyset \wedge \partial U_{t'} \subseteq U_t]$

Proof of claim. $V = U_t \cup R_t$ であり、グラフは全て連結であるので、ある U_t の元 u と、ある R_t の元 v は、辺で結ばれている。

もし $v \notin \partial R_t$ ならば、ある t' $(t' < t)$ で、v は、状態 iv) に入る前に ii)iii) に入っているはずで、すると、$v \in \partial R_{t'}$ となるので、$u \in \partial U_{t'}$ となる。

$v \in \partial R_t$ ならば $u \in \partial U_t$ である。 △

Claim 3 $U_t \neq \emptyset \Rightarrow (\exists t' \ s.t. \ (t < t'))[R_t \subset R_{t'}]$

Proof of claim. R_t の定義より $R_t \subseteq R_{t+1} \subseteq R_{t+2} \subseteq \cdots$. Claim 1,2 より、$\partial U_t \neq \emptyset$ についてある t' について $R_t \cup \partial U_t \subseteq R_{t'}$ △

Claim 4 $(\exists \hat{t})[R_{\hat{t}} = V]$

Proof of claim. Claim 3 より明らか。 △

さて、R_t の元は、leader 以外は、必ず 1 つの親が決まっているので、親 → 子の関係で考えると、各 processor は、in degree 1 で、また leader は、in degree が 0 の連結グラフなので、木になる。 □

Theorem 4.2 NUMBERING $\leq_{\mathrm{m}}$ LEADER

Proof. Lemma 4.1 により生成木を作り、depth-first-search により、各 processor に番号を振る。 □

Theorem 4.3 LEADERは、 $\leq_{\mathrm{m}}$-*complete.*

Proof. Theorem 4.2 により、各頂点は、自分の番号を得るので、隣接する頂点との通信により、各頂点は隣接する頂点の接続情報を得ることができる。よって、この情報を leader に送ることにより、 leader は、ネットワークで与えられたグラフと同型なグラフの表現と port numbering を得ることができる。

よって、 leader は、任意の partial recursive な初期値関数 η を計算でき、その結果を全頂点に放送することにより、任意の初期値関数を計算できる。 □

なお、現在までに以下の階層が明らかになっている。

$$\begin{array}{ccc}
\text{LEADER} \equiv_{\mathrm{m}} \text{NUMBERING} & \equiv_{\mathrm{m}} & \text{EDGE-NUMBERING} \\
\vee_{\mathrm{m}} & & \\
\text{GRAPH-SIZE\& COLORING} & >_{\mathrm{m}} & \text{COLORING} \\
\vee_{\mathrm{m}} & |_{\mathrm{m}} & \vee_{\mathrm{m}} \\
\text{GRAPH-SIZE\& EDGE-COLORING} & >_{\mathrm{m}} & \text{EDGE-COLORING} \\
\vee_{\mathrm{m}} & |_{\mathrm{m}} & \vee_{\mathrm{m}} \\
\text{GRAPH-SIZE} & >_{\mathrm{m}} & \emptyset
\end{array}$$

5 確率的 reduction $\leq_{\mathrm{rm}}$, $\leq_{\mathrm{zpm}}$ について

Proposition 5.1

$$A \leq_{\mathrm{m}} B \;\Rightarrow\; A \leq_{\mathrm{rm}} B,\; A \leq_{\mathrm{zpm}} B$$

$$A \leq_{\mathrm{zpm}} B \;\Rightarrow\; A \leq_{\mathrm{rm}} B$$

Corollary 5.2 LEADERは、 $\leq_{\mathrm{rm}}$, $\leq_{\mathrm{zpm}}$-*complete* である。

Theorem 5.3 *[BIZ89]* LEADER $\leq_{\mathrm{zpm}}$ GRAPH-SIZE

Proof. 入力として、グラフサイズ N をもらった時、以下のアルゴリズムを実行する。

I. 各 processor は、$\mathbf{Z}_{N^3}$ の元を等確率に選ぶ。

II. 選んだ元を list に登録する。

III. 以下を $N-1$ 回行なう。

i) list の内容を隣接する processor 全てに送る。

ii) 送られてきた list を自分の list に追加する。

IV. list の中に N 個の異なる値がない時は、I. へ。

V. 選んだ値が最小の値ならば 1 を出力し、そうでないならば 2 を出力し、停止する。

まず、このアルゴリズムは、$N-1$ 回の繰り返しによって、他の processor の距離が高々 $N-1$ なので、全ての processor の選んだ値が list の要素になる。

よって、後は、$\mathbf{Z}_{N^3}$ の元の中の要素を N 個取った時、各要素が 1 つずつ選ばれる確率が、ある値 p 以上であることを示せれば、平均 $1/p$ 回で停止することがいえる。

N^3 個のものから N 個を unique に取れる確率は、

$$\frac{\begin{pmatrix} N^3 \\ N \end{pmatrix}}{N^{3N}}$$

で表せるが、2 つ取った時に互いに異なる確率は、$1-\frac{1}{N^3}$ で、他の $N-1$ 個と互いに異なる確率は、$(1-\frac{1}{N^3})^{N-1}$ となる。N 個の要素について、それぞれ他の $N-1$ 個と互いに異なる確率は、$(1-\frac{1}{N^3})^{N^2-N}$ となる。

よって、

$$\begin{aligned}
\frac{\binom{N^3}{N}}{N^{3N}} &\geq \left(1-\frac{1}{N^3}\right)^{N^2-N} \\
&= \left(\left(1-\frac{1}{N^3}\right)^{N^3-1}\right)^{\frac{N^2-N}{N^3-1}} \\
&\geq e^{-\frac{N^2-N}{N^3-1}} \\
&= e^{-\frac{N}{N^2+N+1}} \\
&> e^{-\frac{1}{N}}
\end{aligned}$$

よって、示したアルゴリズムは、平均高々 $e^{\frac{1}{N}}$ 回で止まる。 □

Theorem 5.4 GRAPH-SIZE $\not\leq_{\mathrm{rm}}$ COLORING

Proof.

任意の coloring を持った任意のサイズのグラフ上の network で、全 processor に $\frac{1}{2}$ より大きい確率で全頂点数を出力させるアルゴリズムが存在すると仮定する。

すると次のようなグラフ (図 1) の network 上で、$\frac{1}{2}$ より大きい確率で、全 processor は、3 を出力する。

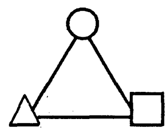

図 1:

このアルゴリズムにおいて、t step まで実行し、停止しなかった時は、空を出力したことにしても、$\frac{1}{2}$ より大きい確率で 3 を出力するような t が存在する。さらに、そ

の t について、アルゴリズムの遷移が離散的なため、3 を出力する確率は、ある $\varepsilon > 0$ という数が存在し、$\frac{1}{2} + \varepsilon$ となる。

さて、次のようなグラフ (図 2) を考える。

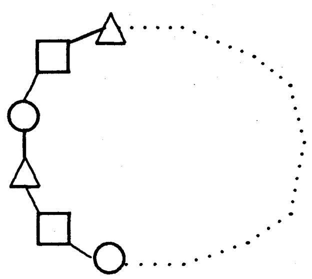

図 2:

このグラフの全頂点数を N とすると、同じ色に着色された一番近い頂点同志が、十分大きい N について、遠い processor からは、t step の内にメッセージが届かないことより、t step 同じ状態にはいっている確率は、N と独立な値 $p > 0$ になる。

すると、各着色された色どうし考えると、どの同じ色で一番近い頂点同志も t step の間の内に同じ状態にいる確率は、$(1-p)^N$ である。

すると、ある n_0 が存在し、$(\forall n \geq n_0)(\exists q)[[1-(1-p)^n > q] \wedge (\frac{1}{2}+\varepsilon)q > \frac{1}{2}]$ となり、その様な n 頂点の上で示したようなグラフについて、この条件より、ある頂点上の processor が、3 を出力する確率は、$\frac{1}{2}$ より大きい。

よって、仮定に矛盾する。 □

Theorem 5.5 COLORING $\leq_{\mathrm{zpm}} \emptyset$

Proof.

アルゴリズム

i) $i \leftarrow 1$

ii) 隣接する頂点を list に登録する。

iii) list から頂点を 1 つ、等確率に選ぶ。

iv) その頂点に 1 か、2 を等確率に選び、送る。

v) その頂点から 1 を送った時 2 、2 を送った時 1 が返って来たら、送った値を a_i とし、選んだ頂点を list から取り除く。そうでなければ、$a_i \leftarrow 0$ とする。

vi) list が空でなければ $i \leftarrow i+1$ とし、iii)へ。

vii) $a_1, \ldots$ を出力し、停止する。

まず、出力した値は、隣接している頂点のものと異なる。それは、u と v が、隣接していれば、ある i のとき、u は v を list から取り除かなければならない。u の出力を $a_i, \ldots, a_m$、v の出力を $b_1, \ldots, b_n$ とすると、$a_i \neq b_i$ である。

次に、平均実行時間は、u が v を選ぶ確率は $\frac{1}{\deg(u)}$ なので、互いに選び合う確率は $\frac{1}{\deg(u)\deg(v)}$. グラフの最大次数を $\deg(G)$ とすると、選び合い、v)で list から取り除き合う確率は $\frac{1}{2}\frac{1}{\deg(G)^2}$.

よって、1 つの頂点について、高々平均 $2\deg(G)$ 回で選ばれることになり、隣接する全ての頂点については、高々平均

$$\sum_{i=1}^{\deg(G)} 2i^2 = \frac{\deg(G)(\deg(G)+1)(2\deg(G)+1)}{3}$$

回で、全て取り除かれる。 □

よって、$\leq_{\mathrm{rm}}, \leq_{\mathrm{zpm}}$ の階層は以下の通りである。

Corollary 5.6

$$\begin{array}{c} \text{GRAPH-SIZE} \equiv \text{LEADER} \equiv \text{NUMBERING} \\ \vee \\ \text{COLORING} \equiv \text{EDGE-COLORING} \equiv \emptyset \end{array}$$

参考文献

[Ang80] D. Angluin. Local and global properties in networks of processors. *Proc. 12th ACM Symp. on Theory of Computing*, pages 82–93, 1980.

[BIZ89] J. Bar-Ilan and Dror Zernik. Random leaders and random spanning trees. *Lecture Notes in Computer Science*, 392:1–12, 1989. Distributed Algorithms.

[GHS87] R. G. Gallager, P. A. Humblet, and P. M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems*, 5(1):66–77, January 1987.

[IR81] A. Itai and M. Rodeh. Symmetry breaking in distributive networks. *Proc. 22nd IEEE Symp. on the Foundations of Computer Science*, pages 150–158, 1981.

[YK89] M. Yamashita and T. Kameda. Electing a leader when processor identity nembers are not distinct. *Lecture Notes in Computer Science*, 392:303–314, 1989. Distributed Algorithms.